Department of Computer Science

Computer Science 177: Modelling and Simulation

MIDTERM EXAMINATION — Spring 2003
SOLUTIONS

1. A Las Vegas casino wants to increase profits and create some excitement by offering ‘‘new and improved’’ forms of gambling. One of their brilliant ideas is to renumber the faces of all of their standard six-sided dice to read:

   "-5", "-3", "-1", "1", "3", "5"

   instead of the numbers "1" through "6". This way, they hope that ordinary games will seem new and exciting, and expert players will get confused by the new markings and not be able to play their winning strategies.

   a. Find the mean and variance for the outcome of rolling one of these modified dice.

      Notice that we are not given a set of samples X_i, but rather need to work with the formula for the probability distribution to find the "true" (i.e., population) mean, μ, and variance, σ^2, rather than the sample mean, $\bar{X}(N)$, and sample variance, $S^2(N)$. Thus:

      $$\mu = \frac{1}{6}\,[(-5) + (-3) + (-1) + 1 + 3 + 5] = 0$$

      $$\sigma^2 = \frac{1}{6}\,[(-5-\mu)^2 + (-3-\mu)^2 + (-1-\mu)^2 + (1-\mu)^2 + (3-\mu)^2 + (5-\mu)^2] = \frac{2}{6}\,[25 + 9 + 1] = \frac{70}{6}$$

   b. Currently, there is one popular game in the player loses his/her bet if they roll ‘‘snake eyes’’ (i.e., a pair of "1"s). What will happen to the casino’s profits from this game if they leave the ‘‘snake eyes’’ rule exactly the same after changing to the new dice, assuming players keep playing the game?

      There will be no change in the profits because we still have only one outcome among the 36 possible outcomes that matches this rule.

2. As the casino’s recently-hired expert on statistics and modelling, you must check the quality of a shipment of new dice to make sure they produce ‘‘fair’’ results. You decide to test ‘‘fairness’’ by having one of your minions conduct the experiment of rolling one die 100 times and recording the results.

   a. Define what we mean by ‘‘fairness’’ in this case, in terms of the desired outcome of those 100 rolls of the dice.

      We want the outcomes of rolling the dice to be uniformly distributed, so each of the 6 possible values comes up just as often, which is pretty obvious. In addition, we also want the outcomes of each roll to be independent of the outcomes of other rolls — we surely would not be happy if the dice came up "snake eyes" on an exact schedule of roll number 1, roll number 37, roll number 73, and so on!

   b. Find the mean and variance for the *sum* of the 100 rolls of the dice, assuming it is ‘‘fair’’.
      [HINT: You shouldn’t need to solve a complicated mathematical formula to answer this question!]

      *Here we use the following facts: (i) the mean of the sum is equal to the sum of the means, whether or not the items are independent, and (ii) the variance of the sum is equal to the sum of the variances, if the items are independent (which we assume to be true here). Therefore the mean value for the sum of 100 rolls of these dice will be 100 * 0 = 0, and the variance of the sum of 100 rolls* *of the dice will be 100 times the variance for one roll, i.e., 100 * (70 / 6) = 1166.666...*

c. Find the mean and variance for the *average* of the 100 rolls of the dice, assuming it is ‘‘fair’’.

Clearly the average value for the mean of 100 rolls of these dice is 0. To find the variance of the mean of 100 rolls, we use the fact that the variance of $a * X$ *is* $a^2 * Var(X)$*, where* $a = 1/100$*, together with the answer to part (b) which shows that the variance of the sum of 100 rolls of the dice (each multiplied by the constant 1/100) will be 100 * (1/100ˆ2 * (70 / 6)) = 7 / 60 = 0.1166666....*

d. Because your minion is very trustworthy, and because of the theoretical properties of the experiment, we can be reasonably sure that the outcome (i.e., the sum and/or mean of the 100 rolls of the dice) be represented by a specific type of statistical distribution. What is this distribution, and why does it apply to this case?

Because we are taking the sum of a large number independent samples from some distribution, the Central Limit Theorem says that the answer will be normally distributed.

e. Based on your minion’s 100 rolls of the die, you calculate that the average value per roll is -0.5. Do you believe the dice are ‘‘fair’’? Justify your answer.

To answer this question, we need to form a confidence interval around the sample mean from our experiment of rolling the dice 100 times, to see if it includes zero. In this case, we know that $\bar{X}(100) = -0.5$ *is given in the problem statement for this question. In addition, we have already calculated the theoretical value for variance of the outcome of averaging 100 rolls of the dice in part (c). Thus, the 95% confidence interval for the true mean for the outcome of averaging 100 tosses of those dice will be:*

$$\bar{X}(100) Z_{0.975} * \sqrt{\frac{\sigma^2}{100}} = -0.5 1.96 * \sqrt{.11666..}$$

Since $\sqrt{.116666}$ *must be greater than 0.3 (since* $0.3^2 = 0.09$ *whereas* $0.4^2 = 0.16$*), the half-width of the confidence interval will be greater than 0.6 and hence includes zero. Thus we cannot reject the hypothesis that* $\mu = 0$*.*

BTW, please note that I specifically used $Z_{0.975}$ *here, rather than* $t_{99,0.975}$*, because the variance in the formula is obtained from the theoretical distribution, rather than calculated from some sample values.*

3. Consider the following code fragments from the CSIM airport shuttle example:

```
    // PASSENGER FRAGMENT
        carlot.reserve();       // join the queue at the car lot
```

```
 2        shuttle_called[CARLOT].set();   // head of queue, so call shuttle
 3        on_carlot.queue();      // wait for shuttle and invitation to board
 4        shuttle_called[CARLOT].clear(); // clear call; next in line will pus
 5        boarded.set();          // tell driver you are in your seat
 6        carlot.release();       // now the next person (if any) is head of q
 7        get_off_now.wait();     // everybody off when shuttle reaches car lo
```

```
      // SHUTTLE FRAGMENT
```

```
 8        while((seats_used < NUM_SEATS) &&
 9        (carlot.num_busy() + carlot.qlength() > 0))
10        {
11           on_carlot.set();// invite one person to board
12           boarded.wait(); // that person is now on board
13           seats_used++;
14           hold(TINY);
15        }
```

Suppose we wish to change the model to include *families travelling together* as a new passenger type. To simplify the problem, we assume that *families block the front of the line, so the shuttle will leave with empty seats rather than allowing others passengers to board.* Modify the two code fragments above to handle families as described above.
[HINT: Let the passenger/family adjust the `seats_used` variable after boarding, and add a new global variable `no_more` to allow the family at the front of the line to send the shuttle away even though it has empty seats if the entire family doesn't fit on board.]

For the passenger fragment, we must create a loop after the family gets to the front of the boarding line to make sure it keeps trying to board a shuttle until it finds one with enough room for the whole family. Otherwise must send the shuttle away before it is full. Thus, we can replace line 3 like this:

```
while (1) {
   on_carlot.queue();// wait for an invitation to board
   if (family_size > (NUM_SEATS - seats_used)){
           // we don't fit
           no_more = true;// shuttle can go away with empty seats
           boarded.set();// wake up driver so he will do it
   }
   else {
           // we do fit
           seats_used += family_size;// get on and occupy our seats
           no_more = false;// tell driver it's his choice what to do next
           boarded.set();// wake up driver so he will do something
           break;
   }
}
```

For the shuttle, we just delete line 13 (since passengers update seat occupancy) and add the following condition to the end of line 9:

```
           && no_more &&
```

4. Programming assignment 1 uses the exponential distribution to represent the time that a light bulb remains operational before it burns out.

a. Show how to generate exponential random variables in C/C++ *without* using the CSIM-18 package.

This is explained on a link from assignment 1.

```
X = - M * log (rand()/(RAND_MAX + 1.0));
```

b. Briefly explain why changing the second lightbulb in a fixture *before* it burns out cannot reduce the maintenance costs in this model.

The key factor is that the life expectancy for light bulbs has an exponential distribution, which is memoryless. That means the remaining lifetime for a working bulb currently in a fixture is the the same as the total lifetime for a new bulb that could be installed in the fixture. Therefore, you get zero extra time, on average, before that light bulb burns out by replacing it before it fails. Since the cost of the replacement is larger than zero (for both materials and labor) and the benefit is exactly zero, this is clearly a waste of time.

Formulas:

$$\bar{X}(n) = \frac{1}{n}\sum_{i=1}^{n} X_i \qquad S^2(n) = \frac{1}{(n-1)}\sum_{i=1}^{n}(X_i - \bar{X}(n))^2$$

$$Prob[abs(\bar{X}(n) - \mu)/\sqrt{\sigma^2/n} < Z_{(1-\alpha/2)}] = 1 - \alpha$$

$$Prob[abs(\bar{X}(n) - \mu)/\sqrt{S^2(n)/n} < t_{(n-1),(1-\alpha/2)}] = 1 - \alpha$$